SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University |
http://www.saujs.sakarya.edu.tr/en/

Title: StationNet: An Algorithm for The Extraction and Visualization of Top-n Correlated Bike Stations in Bike Sharing Systems Big Datasets

Authors: Ahmet Şakir DOKUZ
Recieved: 2020-09-07 13:17:42

Accepted: 2021-01-11 15:43:21

Article Type: Research Article

Volume: 25
Issue: 1
Month: February
Year: 2021
Pages: 275-287

How to cite
Ahmet Şakir DOKUZ; (2021), StationNet: An Algorithm for The Extraction and Visualization of Top-n Correlated Bike Stations in Bike Sharing Systems Big Datasets. Sakarya University Journal of Science, 25(1), 275-287, DOI: https://doi.org/10.16984/saufenbilder.791466
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/58068/791466

Sakarya University Journal of Science 25(1), 275-287, 2021

SAKARYA ÜNİVERSİTESİ

StationNet: An Algorithm for the Extraction and Visualization of Top-n Correlated Bike Stations in Bike Sharing Systems Big Datasets

Ahmet Şakir DOKUZ[1*]

Abstract

Bike sharing systems (BSS) have emerged as an alternative and environmentally friendly transportation tool that provides short-term bike rental to city residents for their close proximity transportation purposes or sports activities. With the emergence and widespread usage of BSS, BSS operators started collecting bike user-related datasets to benefit from it and to increase service quality. Many application areas are present which use BSS big datasets, such as behavioral analyses, urban pattern discovery, and network analysis of bike stations. A bike station network can be defined as a network where bike stations are nodes and the bike trips of users from a station to another station as edges. The extraction of bike station network provides information about which stations are central, which stations have more in- or out-flows, and which regions of the cities are actively used by bike users. However, the extraction of bike station networks is challenging due to the complexity and different characteristics of bike stations, the requirement of new algorithms and new visualization techniques, and the issues related to efficient handling BSS big datasets. In this study, the concept of bike station network extraction in terms of top-n correlated stations is proposed. In particular, the extraction of a bike station network from BSS big datasets are defined and a new algorithm is proposed for extraction of bike station network, and also a new visualization approach that uses common visualization tools is utilized to represent bike station network on a map which would provide more information than a network without a background information. The proposed algorithm and visualization technique are evaluated using one year BSS big dataset. Experimental results show that the proposed algorithm could successfully extract top-n correlated bike station networks and utilized visualization technique is beneficial.

Keywords: top-n correlated bike stations, network visualization, network data mining, BSS big data mining

[*] Corresponding Author: adokuz@ohu.edu.tr

[1] Nigde Omer Halisdemir University, Department of Computer Engineering, Niğde, Turkey, ORCID: https://orcid.org/0000-0002-1775-0954

1. INTRODUCTION

With the rapid increase of fuel consumption in transportation systems led to high carbon emission and the air quality issues in metropolitan cities. To reduce carbon emission and to protect the environment, the city governments and planners seek for alternative transportation systems other than traditional and fuel-based systems. Bike sharing systems (BSS) are one of the clean and environmental friendly transportation systems which both protect environment and provide physical activity for their users [1].

Bike sharing systems (BSS) are becoming one of the popular and essential part of transportation systems for city residents with their environmental friendly nature, fresh outdoor activity possibility, and carbon emissions reducing potential in comparison with traditional transportation systems [2]. BSS locate bike stations at selected parts of the cities where their users can easily reach. The users rent a bike from a station and could drop the bike to another station where the station is close to their desired destination, without requiring to return back to the rental station [3]. Some of the bike stations are highly utilized by bike users due to their central locations and high residential activities. Discovering bike station network is beneficial for BSS operators to better model user activities in terms of in-flows and out-flows for bike stations and to plan new bike stations with respect to high-demanded areas.

Bike station network can be defined as a network where bike stations are nodes and the bike trips of users from a station to another station as edges [4, 5]. Extraction of bike station networks could provide several insights into BSS operators and city planners, such as which stations are central, which stations are origin-preferred or destination-preferred, and which regions of the cities are active in terms of bike user activities.

However, extraction of bike station networks is challenging due to several reasons. First of all, bike station networks are complex and each bike station has its own characteristics. For this reason, new algorithms and methods should be developed for bike station network extraction and visualization. Second, bike sharing big datasets are unstructured and have a high seasonality. Third, bike sharing datasets have a high velocity and are big, and thus, novel approaches should be developed to analyze bike sharing big dataset.

The literature studies related to BSS data mining can be divided into three main categories, i.e. behavioral analysis, spatial and temporal analysis, and network analysis. In behavioral analysis, bike repositioning problem [6-8], effects on bike ridership [9, 10], and trip advisor system [11] are studied. In spatial and temporal analysis, station preference analysis [12-14], urban bike activity analysis [15-17], trip duration prediction [18], and citywide bike usage prediction [19-21] are studied. In network analysis, spatio-temporal graph analysis [23, 25] community discovery [4, 26] and station characteristics extraction [5, 24, 27] are studied.

In bike station network analysis, Calafiore, et al. [22] proposed and implemented a model that is based on closed queuing networks for ToBike BSS system using stations and itineraries between stations as nodes to identify customer arrival rates for each station. Yang, et al. [23] analyzed Nanchang BSS in terms of spatio-temporal and graph-based aspects over a period of a new metro line came into operation and travel behavior changes of before and after the metro line are investigated. Lin, et al. [5] proposed a deep learning-based approach for learning correlations between stations to predict station-level hourly demand. They also performed graph network analysis and visualization to discover more knowledge about station correlations. Oppermann, et al. [24] recorded data from several hundred BSS networks and combined with other data sources and extracted characteristic network metrics from the data. They also designed an online visualization tool to make the dataset publicly available. Zaltz Austwick, et al. [25] used visualization, statistical analyses, and spatial and network analyses to explore BSS usage characteristics of five different cities to investigate features of each city and to uncover similarities between cities. Shi, et al. [26]

proposed an interactive visual analytics system to detect cycling communities of BSS using different community detection algorithms to find bike station communities. Yao, et al. [4] applied detailed network analysis methods to analyze relationships between bike stations of BSS using real-time data of Nanjing public bike system and they resulted that many stations have low usage characteristics and there is a geographical division between highly utilized and low-demand stations. They also discovered top-10 and bottom-10 related stations with ranking stations based on several parameters. Liu, et al. [27] proposed a method which is able to uncover the structure of transportation network and identify bicycle infrastructures to characterize roadways based on different network centrality measures.

In this study, bike station network for BSS is extracted and visualized. For this purpose, first, network extraction algorithm of StationNet algorithm is proposed for extracting most related stations in terms of incoming and outgoing bike activities. Then, two different tools are used as complementary of each other to visualize extracted bike station network. Gephi graph visualization tool is used for construction of nodes and edges, and Google Earth Pro map tool is used for mapping the station network. The experiments are performed for analyzing performances of both StationNet algorithm and the benefit of visualization method.

Main contributions of this study are given as follows:

- The definitions related to bike station network extraction are provided.
- A novel algorithm is proposed for bike station network extraction.
- A new approach is utilized for visualization of extracted bike station network on a map to provide more information about network and background map.
- The proposed method is experimentally evaluated on real-life BSS dataset of Chicago Divvy Bikes.

The rest of this study is organized as follows. Section 2 presents BSS big dataset and preprocessing steps, bike station network extraction and visualization method and proposed algorithm. Section 3 evaluates the experimental results and discussion of the study. Finally, Section 4 outlines the conclusions.

2. MATERIALS AND METHOD

In this section, first the dataset of this study is introduced and the preprocessing steps are explained. Second, basic concepts and definitions of proposed method are expressed. Third, the proposed station network extraction algorithm is presented. Finally, the visualization method of this study is expressed.

2.1. The BSS Dataset

In this study, Divvy Bikes [28] BSS, which operates in Chicago, data are used as a bike sharing systems big dataset. Divvy Bikes shares the bike sharing data since 2013 as a quarter-year dataset. In each quarter-year dataset, three months of BSS activity data are present. 4 quarter-year dataset from 2018 are selected for this study, and the selected dataset is from 1 January 2018 to 31 December 2018. The total number of BSS activities in selected dataset are 3.603.082 for the year of 2018. Figure 1 presents the number of BSS activities for the selected dataset. As can be seen in the figure, Q1 and Q4 have less number of activities because these quarters contain winter and autumn months, and thus bike usages decrease because of the weather conditions. The majority of the BSS activities occurred at Q2 and Q3, which are within spring and summer seasons.

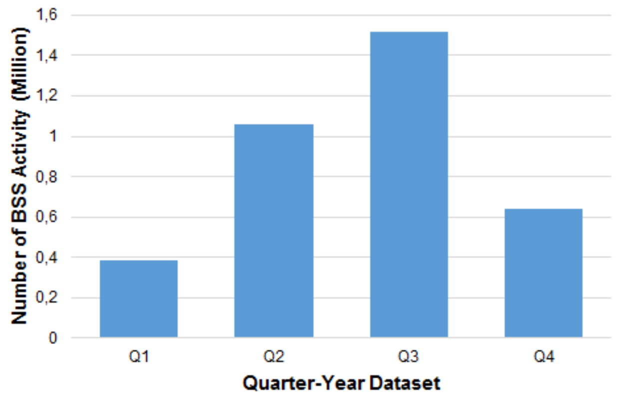


Figure 1 Number of BSS activities for each quarter-year dataset

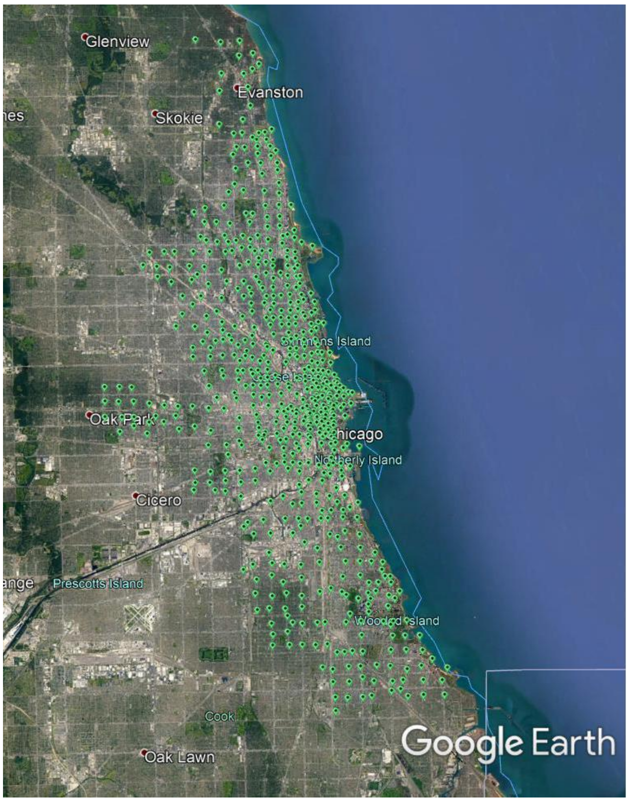


Figure 2 The distribution of all stations of Divvy Bikes [28]

There are 611 bike stations present in Divvy Bikes BSS. Figure 2 presents the distribution of all bike stations in Divvy Bikes on a map. As can be seen in Figure 2, the stations are distributed among Chicago and are clustered around the city center. Divvy Bikes [28] BSS provides latitude and longitude information of stations which are obtained as a file and used in this study.

The original dataset contains 12 columns that store information about each individual bike trips. For this study, only 3 of these columns are used, i.e., start station, end station, and trip duration.

2.1.1. Preprocessing

As preprocessing steps, first of all, quotation marks and commas in trip duration information are removed from the dataset due to achieve numerical trip duration information. Then, trip duration of each trip is checked whether the duration last more than 12 hours using a similar strategy with [4]. If the trip lasts more than 12 hours, then these trips are removed from the dataset, because no real bike trips could take such long time. After removing longer trips, the final dataset has the number of BSS activities of 3.599.798.

2.2. Basic Concepts

In this section, basic concepts and definitions related to the proposed bike station network extraction method are presented.

Definition 1. A **bike station** *bs* is the station that a user could rent or return a bike in BSS.

Definition 2. Given two bike stations *bs1* and *bs2*, a **bike trip** *bt* is a directed trip from *bs1* to *bs2* in BSS dataset.

For example, if a user rent a bike from *bs1* and return the bike to *bs2*, then trip *bt* is constructed as *bs1→bs2*.

Definition 3. Given a bike trip *bt*, an **incoming station** is the station that *bt* started from and ended at the selected bike station.

For example, if a *bt* constructed as *bs1→bs2* and our selected bike station is *bs2*, then incoming station in *bt* is *bs1*.

Definition 4. Given a bike trip *bt*, an **outgoing station** is the station that *bt* is ended and started from the selected bike station.

For example, if a *bt* constructed as *bs1→bs2* and our selected bike station is *bs1*, then outgoing station in *bt* is *bs2*.

Definition 5. Given a list of bike trips *btlist* and a bike station *bs*, **top-n incoming station** for *bs* is the stations that have n-highest number of trips from an incoming station to *bs*.

Definition 6. Given a list of bike trips *btlist* and a bike station *bs*, **top-n outgoing station** for *bs* is the stations that have n-highest number of trips from *bs* to an outgoing station.

Definition 7. Given a list of bike stations *bslist* and top-n incoming and outgoing stations lists of each bike station *bs* in *bslist*, a **station network** of *bslist* is top-n incoming and outgoing stations for each station *bs* in *bslist*.

2.3. Station Network Extraction

In this section, the proposed Station Network (StationNet) algorithm, which is developed using the basic definitions in Section 3.2, is presented. The algorithm takes BSS trip dataset as input and extracts the station network with respect to trip information in the dataset. The benefit of StationNet algorithm is that it can extract most related stations using bike trips in BSS datasets and flow of the bike users among the city bike stations could be observed. Algorithm 1 presents the pseudo code of StationNet algorithm.

Algorithm 1. StationNet Algorithm

Inputs:
D: The BSS trip dataset
S: The bike stations dataset
top-n: Top-n mostly related stations of each station in *S*

Output:
Network: The station network with top-n most correlated stations.

Algorithm:

```
1.  Network = [][][][]
2.  dataset = preprocess-dataset(D)
3.  bsList = readDataset(S)
4.  for each bike station bs in bsList
5.    incomingStation = [], outgoingStation = []
6.    for each bike trip bt in dataset
7.      if bs == bt.origin and bs == bt.destination then
8.        continue;
9.      end if
10.     if bs == bt.origin then
11.       outgoingStation.add(bt.destination)
12.     else if bs == bt.destination then
13.       incomingStation.add(bt.origin)
14.     end if
15.   end for
16.   [topNIncomingStations, topNIncomingWeights
      topNOutgoingStations, topNOutgoingWeights] =
      extract-top-n-stations(bsList, incomingStation,
      outgoingStation, top-n)
17.   Network.add(bs, topNIncomingStations,
      topNIncomingWeights, topNOutgoingStations,
      topNOutgoingWeights)
18. end for
19. return Network
```

As can be seen in Algorithm 1, the StationNet algorithm takes BSS trip and station datasets and *top-n* value as input and extracts station network for given datasets. At the first step, network matrix initialization is performed. At step 2, the dataset is preprocessed as presented in Section 2.1.1. At step 3, station dataset is read into *bsList* array. At steps between 4 and 18, the station network extraction is performed for each station. At steps between 6 and 15, each trip in the dataset is traversed and incoming and outgoing stations for selected station are extracted. At steps between 7 and 9, if selected station *bs* is at both start and end station of a bike trip, the trip is ignored, because the trip information do not add any value to station network. At steps between 10 and 14, if the trip has selected station as origin, then destination station of the trip is included in outgoing list, and if the trip has selected station as destination, then origin station of the trip is included in incoming list. At step 16, based on incoming and outgoing stations list of selected station, top-n incoming and outgoing stations are extracted. At step 17, the extracted top-n incoming and outgoing stations and their frequencies are added to the station network. Finally, at step 19, the constructed station network is returned as the output of StationNet algorithm.

Algorithm 2. *extract-top-n-stations* function

Inputs:
bsList: The bike stations list
incomingStation: Incoming stations list
outgoingStation: Outgoing stations list
top-n: Top-n mostly related stations

Output:
[topNIncomingStations, topNOutgoingStations]: The top-n most correlated stations with respect to given station.

Algorithm:

```
1. incomingFreqArray[], outgoingFreqArray[]
2. for each bike station bs in bsList
3.   incomingFreq = incomingStation.frequency(bs)
4.   outgoingFreq = outgoingStation.frequency(bs)
5.   incomingFreqArray.add(incomingFreq)
6.   outgoingFreqArray.add(outgoingFreq)
7. end for
8. [topNIncomingStations, topNOutgoingStations] =
   find-top-n-frequency(bsList, incomingFreqArray,
   outgoingFreqArray)
9. return [topNIncomingStations,
   topNOutgoingStations]
```

Algorithm 2 takes bike station list *bsList*, incoming station list *incomingStation*, outgoing station list *outgoingStation* and *top-n* value as input and returns top-n incoming and outgoing stations. At step between 2 and 7, all stations are traversed and incoming and outgoing frequencies of stations are extracted and saved to an array. Then, at step 8, the stations that have top-n highest frequencies with respect to selected station are determined. Finally, at step 9, the algorithm returns top-n incoming and outgoing stations.

2.3.1. Analysis of StationNet Algorithm

In this section, CPU utilization, RAM usage and complexity of StationNet algorithm are investigated. CPU utilizes 5 threads for running StationNet algorithm. For 4 quarter year dataset and for *top-n* value of 3, the algorithm execution time on CPU is 242 seconds. Detailed CPU time usages with respect to algorithm parameters are provided in experimental evaluation.

RAM usage of StationNet algorithm at the highest level is approximately 3.4 GB and nearly 80% of the RAM is utilized for arrays.

Algorithm complexity of StationNet algorithm is $O(n^2)$ because the algorithm contains two nested loops.

2.4. Station Network Visualization

In this section, the visualization of bike station network which is extracted by StationNet algorithm is presented. For visualization purposes, graph visualization and mapping tools are preferred which are used for visualizing and mapping complete station network.

In this study, for visualization of bike station network, open-source graph visualization tool of Gephi [29] is preferred due to its several benefits, such as geographical representation and edge weighting. Although, Gephi has geographical representation and map visualization of graphs, the map visualization ability is not sufficient at desired level. So, free version of Google Earth Pro [30] is also used for mapping the graph of the station network that is outputted from Gephi. The flow of visualization using Gephi and Google Earth Pro are presented in Figure 3.

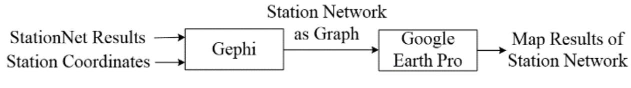
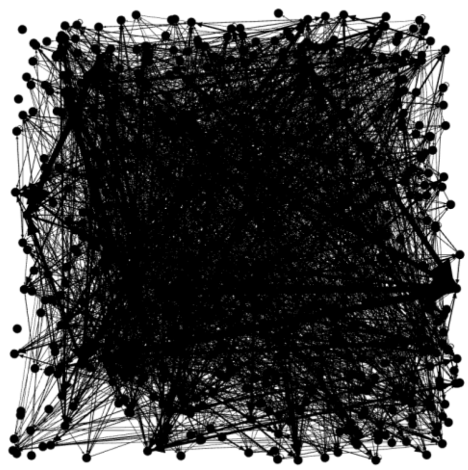


Figure 3 The visualization flow of the station network

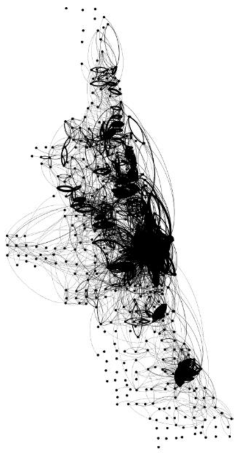
As can be seen in Figure 3, first of all, StationNet algorithm results and station coordinate information are given to Gephi and station network is extracted as a graph. Then, station network graph is given to Google Earth Pro and Google Earth Pro maps the graph onto the earth map with the connections between each station.

At the visualization step, Gephi treated each station as nodes and each connection between stations as edges. By this way, a graph is constructed. Also, the weights of each edge are given to Gephi as the number of trips originated from a station to the other. Gephi has the ability to use geographical coordinates and locate each node in their corresponding location. For this purpose, the coordinates of each node (or station) are given as input to Gephi to correctly locate each node as shown in Figure 3.

Two different visualizations for a station network graph is presented in Figure 4. In Figure 4 (a), the default visualization is used, and there is no geographical property is used. In Figure 4 (b), geographical properties of stations are used for locating each node. As can be seen in Figure 4 (b), Gephi successfully visualizes station network, however, a map is missing at the background.



(a)



(b)

Figure 4 (a) The default and (b) geographical visualization of station network graph by Gephi

Gephi can extract the graphs to various output formats. One of these formats is KML file which is the format of Google Earth Pro. For this reason, the station network graph is extracted as KML file and given to Google Earth Pro. Figure 5 presents the output of Google Earth Pro with the station network that Gephi extracted. As can be seen in Figure 5, Google Earth Pro could construct the station network with their edges on a map at background.

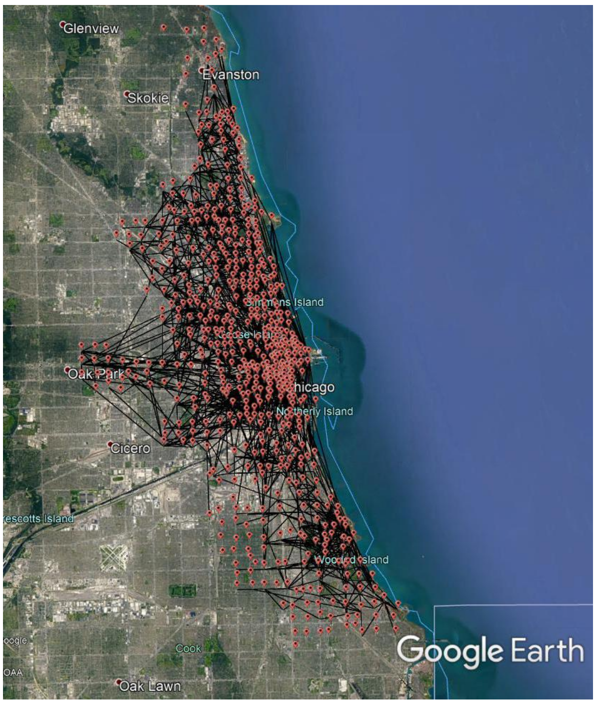


Figure 5 The mapping of station network graph by Google Earth Pro

3. RESULTS AND DISCUSSION

In this section, the experimental results of the proposed station network extraction method are presented. First, the proposed StationNet algorithm is evaluated whether it is a scalable algorithm. Then, network analysis of top-n is performed with different values that affect network results. Finally, the results of StationNet algorithm are evaluated. The experimental setup of this study is presented in Figure 6.

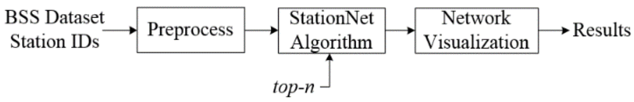


Figure 6 Experimental setup of this study

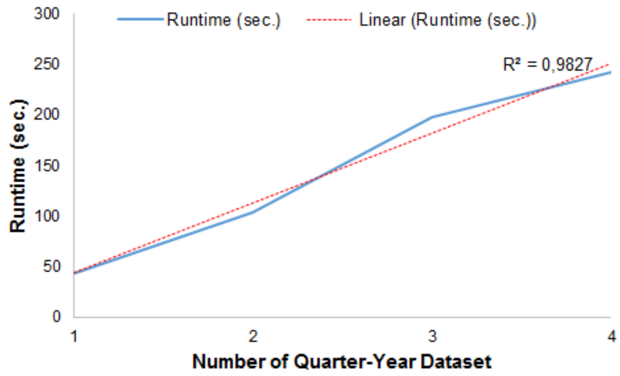
The experiments are performed on a computer with Intel Core i7 CPU with 3.40 GHz, and 8 GB of RAM. The algorithm is written in Java programming language.

3.1. Scalability Analysis

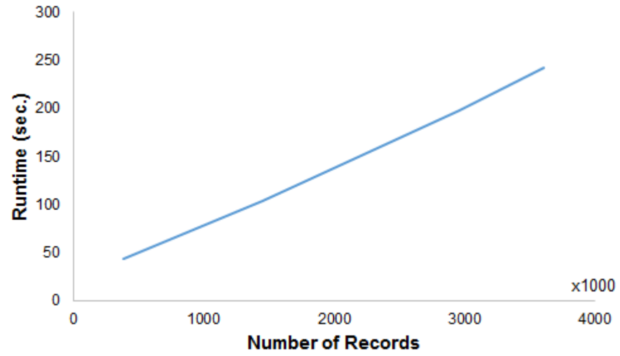
In this experiment, StationNet algorithm is evaluated whether it is scalable with the increase of the number of quarter-year dataset and the increase of the value of *top-n*. The number of quarter-year dataset is selected as 1, 2, 3, and 4 while *top-n* value is 3. The employed quarter year datasets and record counts for each number of dataset are presented in Table 1. The value of *top-n* is increased from 2 to 20, by 2 while quarter-year dataset is 2. The experimental results are presented in Figures 7 and 8.

Table 1
Dataset description

Number of Dataset	Quarters	Record Count
1	Q1	387146
2	Q1 and Q2	1446828
3	Q1, Q2 and Q3	2960399
4	Q1, Q2, Q3 and Q4	3603086



(a)



(b)

Figure 7 Runtime of StationNet algorithm for a) different number of quarter-year dataset, and b) number of records

As can be seen in Figure 7 a) and b), execution time of StationNet algorithm increases with the increase of the number of quarter-year dataset and the number of records. The increase amount is highest for 3 quarter-year dataset, because 3rd quarter contains summer months and the bike usage counts dramatically increase in this season. As can be seen in Figure 7 a), the increase trend of StationNet algorithm with respect to the number of quarter-year dataset could be seen as scalable because the execution time graphic is compatible with linear trend line with R^2 value of 0.9827.

As can be seen in Figure 8, the execution time of the StationNet algorithm increases with the increase of *top-n* value, as expected. When *top-n* value is increased, the algorithm tries to find more stations related to one station in the dataset, and thus the execution time increases. The increase trend of the algorithm with respect to *top-n* value is compatible with linear trend line with R^2 value of 0.9915, because, the execution time of the algorithm increases linearly with the increase of *top-n* value.

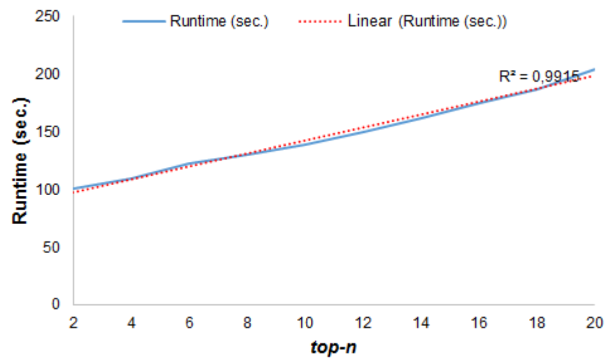


Figure 8 Runtime of StationNet algorithm for different *top-n* value

As a result of this experiment, StationNet algorithm could be seen as a scalable and time-efficient algorithm with respect to given dataset, because it could successfully handle the increase of both the dataset and *top-n* value linearly as presented in Figures 7 and 8. Although, the algorithm complexity is $O(n^2)$, one of the nested loops is constant, the loop which traverses all stations. For this reason, the increase in the quarter year dataset leads to the linear increase of the execution time of the algorithm.

3.2. Network Analysis

In this experiment network analyses for different *top-n* values are performed. All quarters of the dataset, i.e. Q1, Q2, Q3 and Q4, are used in this experiment with a total of nearly 3,6 million records. *top-n* value is selected as 1, 2, 3, 5, 10, and 20 and the results of StationNet algorithm is evaluated for each *top-n* value. Table 2 presents statistics and network properties for each *top-n* value. Also, Figure 9 presents the station network for each *top-n* value.

Table 2
Network properties for different *top-n* values

	***top-n* = 1**	***top-n* = 2**	***top-n* = 3**	***top-n* = 5**	***top-n* = 10**	***top-n* = 20**
Number of Nodes	604	604	604	604	604	604
Number of Edges	1027	1962	2871	4673	8924	14803
Edges / Nodes	1.700	3.248	4.753	7.737	14.775	24.508

Graph Density	0.003	0.005	0.008	0.013	0.024	0.039
Modularity	0.843	0.779	0.746	0.706	0.64	0.568
Average Degree	1.681	3.206	4.692	7.635	14.581	24.031
Network Diameter	29	25	17	11	9	6
Average Path Length	9.014	8.302	5.908	4.489	3.431	2.829
Average Clustering Coefficient	0.069	0.274	0.294	0.3	0.31	0.307

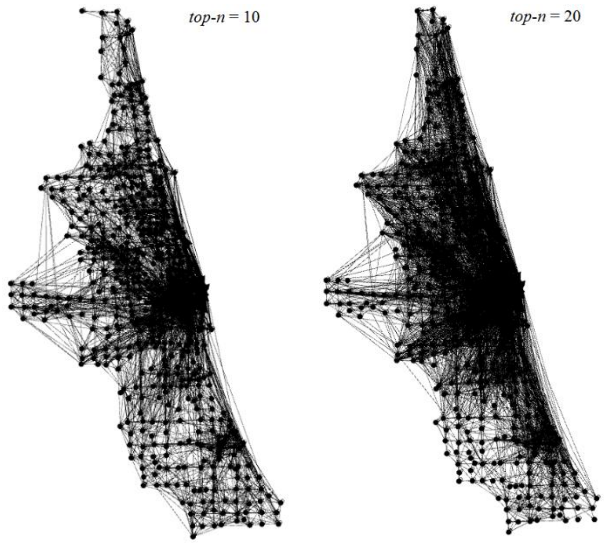


Figure 9 The station network visualization for different *top-n* values

As can be seen in Table 2, the number of nodes is constant, while the number of edges increase with the increase of *top-n* value. However, the increase ratio is not linear, which can be observed in Edges / Nodes ratio, due to the reason that some stations do not have enough number of correlated stations. Graph density, average degree, and average clustering coefficient increases with the increase of *top-n* value because the network becomes more complex with the increase of *top-n* value. Contrarily, modularity, network diameter, and average path length decreases with the increase of *top-n* value, because these properties are related with connectedness of the networks.

As can be seen in Figure 9, the station network gets more complex with the increase of the *top-n* value. More edges are constructed between stations due to the increase of *top-n* value. However, when *top-n* value is increased unnecessarily, the network becomes non-functional and nearly all nodes are connected together. When all networks are investigated, the central of Chicago have high density of edges compared to other locations.

3.3. Evaluation of the Results

In this experiment, the results of StationNet algorithm are evaluated. The number of quarter-year dataset is selected as 4 and *top-n* value is selected as 5. First, the station network is visualized with the edges on a map. Second, the most central 10 stations that have highest incoming and outgoing edges are extracted and presented on a map.

As can be seen in Figure 10, the station network is clustered around the central of Chicago and near the bay area. Also, local clusters are observed near Oak Park, Ravenswood, Washington Park, and Clybourn train station.

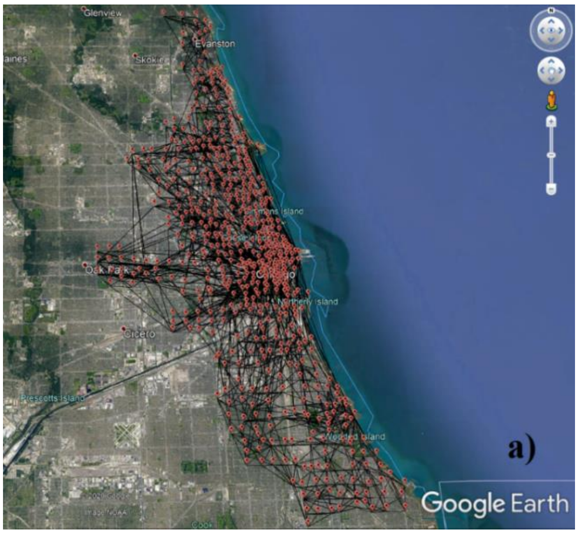


Figure 10 The station network shown on a map

Table 3 and Figure 11 presents the most central 10 stations that have highest number of edges. As can be seen in Table 3, some stations have higher outgoing edges, and other stations have higher incoming edges. For example, station 192 have 75 outgoing edges and 65 incoming edges. This means that station 192 is more preferred as an origin station. Also, the total edge counts of these 10 stations show that these stations are strongly linked with other stations in the network.

Table 3
The most central 10 stations that have the highest outgoing and incoming edges

#	Station ID	Outgoing	Incoming	Total
1	192	75	65	140
2	77	65	69	134
3	91	64	67	131
4	35	46	51	97
5	344	43	43	86
6	69	36	34	70
7	133	28	39	67
8	58	36	30	66
9	239	29	31	60
10	174	26	33	59

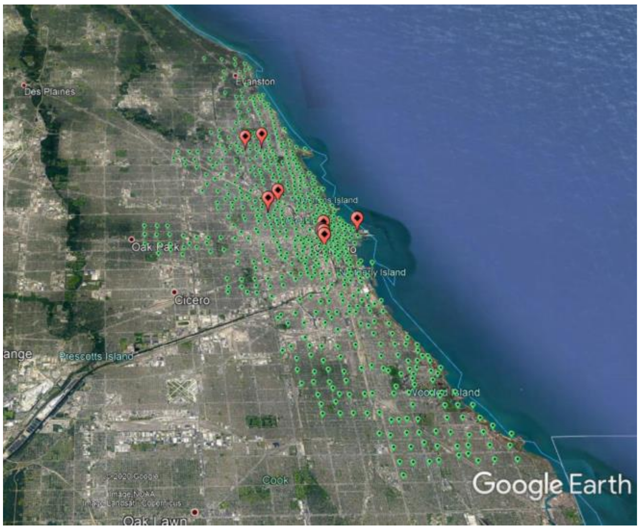


Figure 11 The most central 10 stations (red points) and other stations (green points) on a map

In Figure 11, the most central stations are presented as red points and other stations are presented as green points. As can be seen in the figure, the most central stations are within the center of the Chicago city and these stations are in between other stations. Five of these stations are clustered at the city center. Other stations are distributed among the center and northern parts of the city. As a result, the bike users tend to visit these stations more frequent than other stations for starting or finishing their bike trips due to their various reasons.

4. CONCLUSIONS

BSS provide city residents a new transportation system while providing them a way to do sports and reach their desired destinations in the cities at the same time. The widespread use of BSS resulted a big dataset and a new knowledge platform about the citizen behavior in cities. The researchers utilized BSS datasets for various purposes, including behavioral analysis of city residents, urban pattern analysis of BSS users, and extraction of network analysis between bike stations based on the usage characteristics and utilization of these stations.

This study focuses on developing a novel algorithm for extraction of bike station networks on BSS big datasets. The proposed algorithm is designed to handle and to effectively run on BSS big datasets. Also, a new approach is utilized to visualize extracted bike station network which use Gephi and Google Earth Pro visualization and mapping tools sequentially to visualize network interaction of bike stations on a map. The proposed bike station network extraction and visualization method is experimentally evaluated on Chicago Divvy Bikes BSS big dataset. The experimental results show that the proposed bike station network extraction algorithm of StationNet algorithm is scalable with the increase of the dataset size for given BSS big dataset and the algorithm could successfully extract bike station network of given BSS big dataset. Also, visualization tools are seen as beneficial of understanding the network structure of the StationNet algorithm.

A limitation of this study is that, this study does not exclude low profile stations, which brings extra complexity to the algorithm. Also, the mapping of all station network leads to complicated visualization and each station could not be observed separately.

For the future studies, the community detection from the results of StationNet algorithm could be performed. Also, spatial and temporal analyses of bike station networks could be applied on BSS big datasets. The extraction and visualization of bike station networks in BSS datasets could be performed using big data technologies, such as Hadoop MapReduce and cloud computing resources.

Acknowledgments

The author wishes to thank the editor and reviewers for their valuable comments about the study. The author also wishes to thank to Divvy Bikes BSS operator which publicly provide the dataset.

Funding

The author has no received any financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] Y. Zhang and Z. Mi, "Environmental benefits of bike sharing: A big data-based analysis," Applied Energy, vol. 220, pp. 296-301, 2018/06/15/ 2018.

[2] E. Eren and V. E. Uz, "A review on bike-sharing: The factors affecting bike-sharing demand," Sustainable Cities and Society, p. 101882, 2019/10/15/ 2019.

[3] S. A. Shaheen, S. Guzman, and H. Zhang, "Bikesharing in Europe, the Americas, and Asia:Past, Present, and Future," Transportation Research Record, vol. 2143, pp. 159-167, 2010.

[4] Y. Yao, Y. Zhang, L. Tian, N. Zhou, Z. Li, and M. Wang, "Analysis of Network Structure of Urban Bike-Sharing System: A Case Study Based on Real-Time Data of a Public Bicycle System," Sustainability, vol. 11, p. 5425, 2019.

[5] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," Transportation Research Part C: Emerging Technologies, vol. 97, pp. 258-276, 2018/12/01/ 2018.

[6] M. Dell'Amico, M. Iori, S. Novellani, and A. Subramanian, "The Bike sharing Rebalancing Problem with Stochastic Demands," Transportation Research Part B: Methodological, vol. 118, pp. 362-380, 2018/12/01/ 2018.

[7] G. Erdoğan, M. Battarra, and R. Wolfler Calvo, "An exact algorithm for the static rebalancing problem arising in bicycle sharing systems," European Journal of Operational Research, vol. 245, pp. 667-679, 2015/09/16/ 2015.

[8] S. Ban and K. H. Hyun, "Curvature-based distribution algorithm: rebalancing bike sharing system with agent-based simulation," Journal of Visualization, vol. 22, pp. 587-607, 2019/06/01 2019.

[9] R. A. Rixey, "Station-Level Forecasting of Bikesharing Ridership: Station Network Effects in Three U.S. Systems," Transportation Research Record, vol. 2387, pp. 46-55, 2013/01/01 2013.

[10] Y. Xing, K. Wang, and J. J. Lu, "Exploring travel patterns and trip purposes of dockless bike-sharing by analyzing massive bike-sharing data in Shanghai, China," Journal of Transport Geography, vol. 87, p. 102787, 2020/07/01/ 2020.

[11] P. Cheng, J. Hu, Z. Yang, Y. Shu, and J. Chen, "Utilization-Aware Trip Advisor in Bike-Sharing Systems Based on User Behavior Analysis," IEEE Transactions on Knowledge and Data Engineering, vol. 31, pp. 1822-1835, 2019.

[12] A. Faghih-Imani and N. Eluru, "Analysing bicycle-sharing system user destination choice preferences: Chicago's Divvy system," Journal of Transport Geography, vol. 44, pp. 53-64, 2015/04/01/ 2015.

[13] M. Hyland, Z. Hong, H. K. R. d. F. Pinto, and Y. Chen, "Hybrid cluster-regression approach to model bikeshare station usage," Transportation Research Part A: Policy and Practice, vol. 115, pp. 71-89, 2018/09/01/ 2018.

[14] A. S. Dokuz, "Station Preference Analysis of Users in Bike Sharing Systems Big Datasets," European Journal of Science and Technology, vol. 2020 Special Issue, pp. 591-597, 1 April 2020 2020.

[15] P. Jiménez, M. Nogal, B. Caulfield, and F. Pilla, "Perceptually important points of mobility patterns to characterise bike sharing systems: The Dublin case," Journal of Transport Geography, vol. 54, pp. 228-239, 2016/06/01/ 2016.

[16] J. Wergin and R. Buehler, "Where Do Bikeshare Bikes Actually Go?: Analysis of Capital Bikeshare Trips with GPS Data," Transportation Research Record, vol. 2662, pp. 12-21, 2017/01/01 2017.

[17] W. Li, S. Wang, X. Zhang, Q. Jia, and Y. Tian, "Understanding intra-urban human mobility through an exploratory spatiotemporal analysis of bike-sharing trajectories," International Journal of Geographical Information Science, vol. 34, pp. 2451-2474, 2020/12/01 2020.

[18] V. E. Sathishkumar, J. Park, and Y. Cho, "Seoul bike trip duration prediction using data mining techniques," IET Intelligent Transport Systems, vol. 14, pp. 1465-1474, 2020.

[19] Y. Li and Y. Zheng, "Citywide Bike Usage Prediction in a Bike-Sharing System," IEEE Transactions on Knowledge and Data Engineering, pp. 1-1, 2019.

[20] X. Ma, Y. Ji, Y. Yuan, N. Van Oort, Y. Jin, and S. Hoogendoorn, "A comparison in travel patterns and determinants of user demand between docked and dockless bike-sharing systems using multi-sourced data," Transportation Research Part A: Policy and Practice, vol. 139, pp. 148-173, 2020/09/01/ 2020.

[21] N. Boufidis, A. Nikiforiadis, K. Chrysostomou, and G. Aifadopoulou, "Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators," Transportation Research Procedia, vol. 47, pp. 51-58, 2020/01/01/ 2020.

[22] G. C. Calafiore, F. Portigliotti, and A. Rizzo, "A Network Model for an Urban Bike Sharing System" IFAC-PapersOnLine, vol. 50, pp. 15633-15638, 2017/07/01/ 2017.
[23] Y. Yang, A. Heppenstall, A. Turner, and A. Comber, "A spatiotemporal and graph-based analysis of dockless bike sharing patterns to understand urban flows over the last mile," Computers, Environment and Urban Systems, vol. 77, p. 101361, 2019/09/01/ 2019.
[24] M. Oppermann, T. Möller, and M. Sedlmair, "Bike Sharing Atlas: Visual Analysis of Bike-Sharing Networks," International Journal of Transportation, vol. 6, pp. 1-14, 2018.
[25] M. Zaltz Austwick, O. O'Brien, E. Strano, and M. Viana, "The Structure of Spatial Networks and Communities in Bicycle Sharing Systems," PLOS ONE, vol. 8, p. e74685, 2013.
[26] X. Shi, Y. Wang, F. Lv, W. Liu, D. Seng, and F. Lin, "Finding communities in bicycle sharing system," Journal of Visualization, vol. 22, pp. 1177-1192, 2019/12/01 2019.
[27] X. C. Liu, J. Taylor, R. J. Porter, and R. Wei, "Using trajectory data to explore roadway characterization for bikeshare network," Journal of Intelligent Transportation Systems, vol. 22, pp. 530-546, 2018/11/02 2018.
[28] D. Bikes. (2020). Divvy Bike Sharing Dataset. Available: https://www.divvybikes.com/
[29] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," presented at the International AAAI Conference on Weblogs and Social Media, San Jose, California, USA, 2009.
[30] Google. (2020, 02.04.2020). Google Earth Pro. Available: https://www.google.com.tr/intl/tr/earth/